Imagination
TECHNOLOGIES

Augmented Reality on Mobile Platforms

POWERVR Developer Technology

August 2010

Imagination Overview

- **Leading Semiconductor IP Supplier**
  - POWERVR™ graphics, video, display processing
  - ENSIGMA™ receivers and communications processors
  - META™ processors – SoC centric real-time, DSP, Linux
  - Licensees: Leading Semis and OEMs
  - #4 Silicon Design IP provider *

- **Innovative Consumer Product Manufacturer**
  - PURE digital radio, internet connected audio

- **Established technology powerhouse**
  - Founded:1985
  - Listed:1994-London Stock Exchange:IMG
  - Employees: more than 620 worldwide

** Source: Gartner IP Suppliers Report, March 2009*

UK Headquarters

R&D

Sales

POWERVR Graphics

- **Unique Tile Based Deferred Rendering architecture (TBDR)**
  - Enhances performance
  - Reduces power consumption
- **MBX Series**
  - Fixed-function graphics acceleration
  - Widely adopted in mobile devices
- **SGX Series**
  - Programmable, shader-based graphics acceleration
  - Newer technology already available in 100+ platforms


Apple iPhone 3GS


Samsung Wave


Motorola Droid/Milestone


Samsung i8910HD


Sony Ericsson Satio


Vivaz


HTC Qilin (Dopod T8388)


Nokia N900


Vodafone 360 H1 (Samsung)


Emblaze Else


Palm Pre

POWERVR Graphics

Imagination

Dell
Mini 10

Acer
Aspire One AO751H

Nokia
Booklet 3G

Sony
Vaio X-series

Sony
Vaio P-series

Asus
S121

Fujitsu
UH900

Archos 9

Sharp
Willcomm D4

Clarion
Mind

Archos 5

Apple
iTouch 32GB

Asus
R50A

Imagination
TECHNOLOGIES

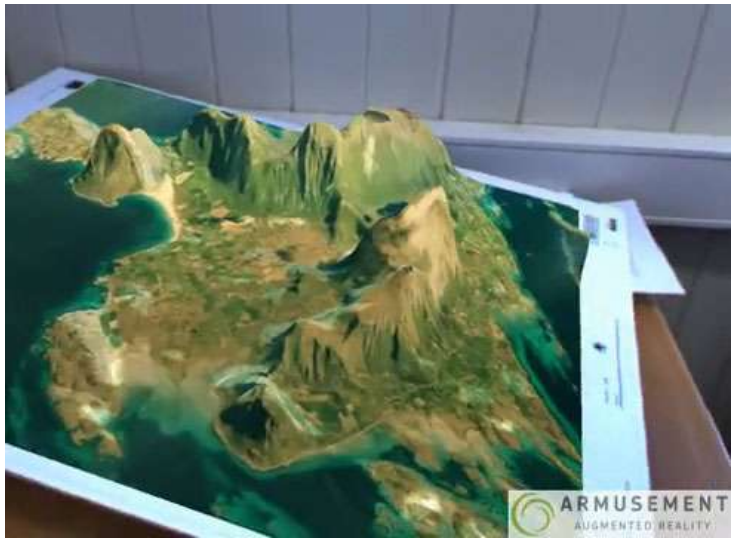
Augmented Reality

What is Augmented Reality?

- **Augmenting a view of a real scene with computer-generated images**
  - Typically reality is captured through a video stream from a camera
  - Images are rendered over the video before display to the user
  - If the video is analysed then objects in the world can be tracked and mixed with the video more realistically
    - Position, orientation and distance information is retrieved

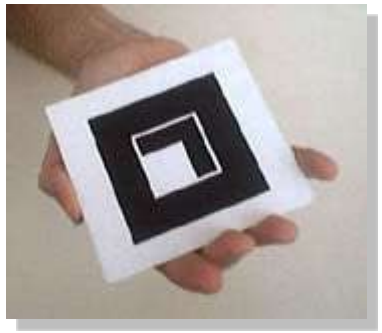
- **Applying captured images to 3D objects**
  - Use images from a camera or video source as textures
    - User interfaces
    - Refraction effects
    - Post processing

- **Example applications**
  - Games
  - Navigation
  - Medical applications (i.e. assisted surgery)



Typical AR Procedure

- **Frame is captured by the camera**
- **This may be passed to some image processing algorithms that look for features in the frame**
  - Often these are artificial markers that are easier for the analysis to find than normal features
  - Sometimes other inputs such as GPS position, accelerometer or compass information is used
- **The application then decides how this affects where and what to render in terms of additional graphics**
- **The graphics are rendered over the camera frame in a position that gives the illusion of the computer-generated content being part of the camera image**



Find AR Marker

AR on Desktop

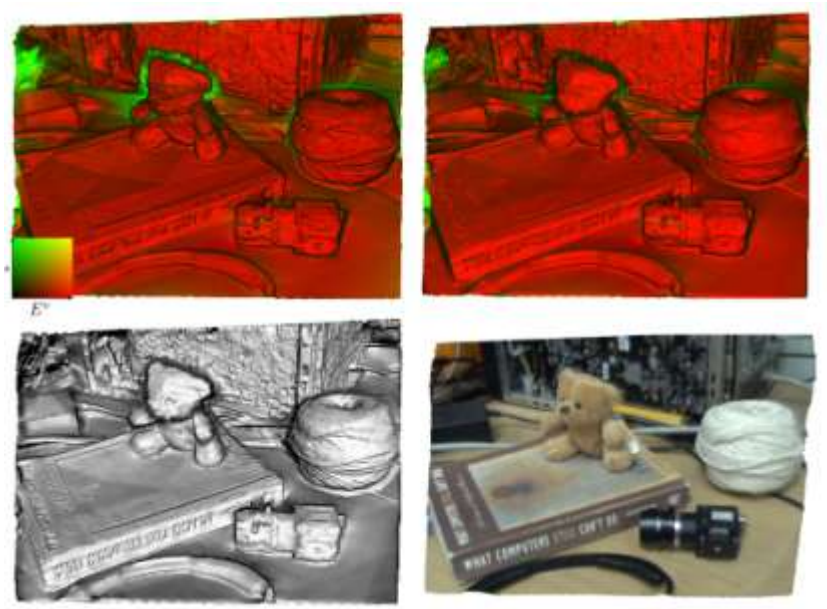
- **Benefits**
  - Fast hardware to process and analyse input images
  - Fast hardware to render graphics
  - Runs from a mains power supply (not limited by finite battery life)

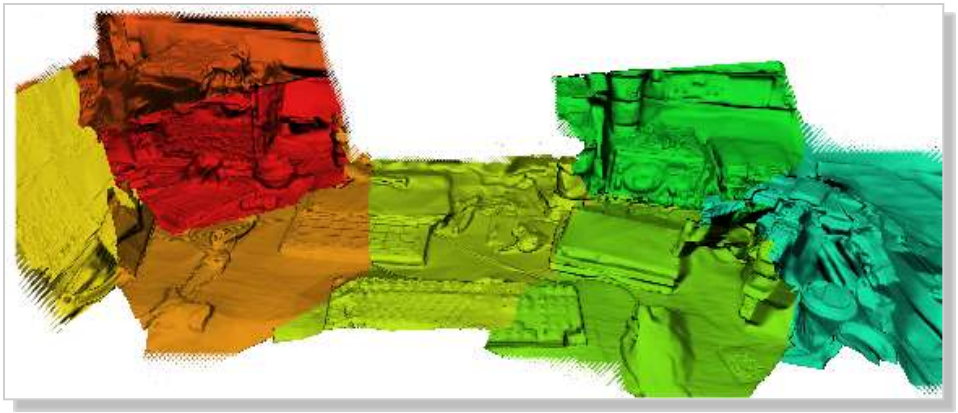
- **Drawbacks**
  - Difficult to transport
  - Clumsy to handle
  - Requires a local power supply

- **Examples**
  - Holding a marker from a magazine up to a webcam on a desktop PC allows extra content to be viewed
  - Virtual dressing rooms
    - Fixed camera
    - Sometimes no image tracking
  - Sports events
    - NFL first down marker

Research work: SLAM
AR on Desktops

- **What is SLAM?**
  - Real-time three dimensional reconstruction of scenes viewed by a single camera
  - System has a real understanding of its environment without relying on artificial aids like markers
- **Benefits**
  - Understand the position of static objects in the real world
  - Works within the restriction of a single camera input
  - No markers required
- **Drawbacks**
  - Requires high end desktop hardware to process in real-time at the moment





Material courtesy of the Imperial College of London. Available : http://www.doc.ic.ac.uk/~ajd/

AR on mobile

- **Benefits**
  - Highly transportable
  - Many modern devices, such as phones, have built in cameras
  - Large installation base of potential users
- **Drawbacks**
  - Battery constraints
  - Less processing power than desktop
    - Image analysis
    - Rendering
- **Example**

ARf virtual pet (AAEL, Georgia Tech)

AR on mobile

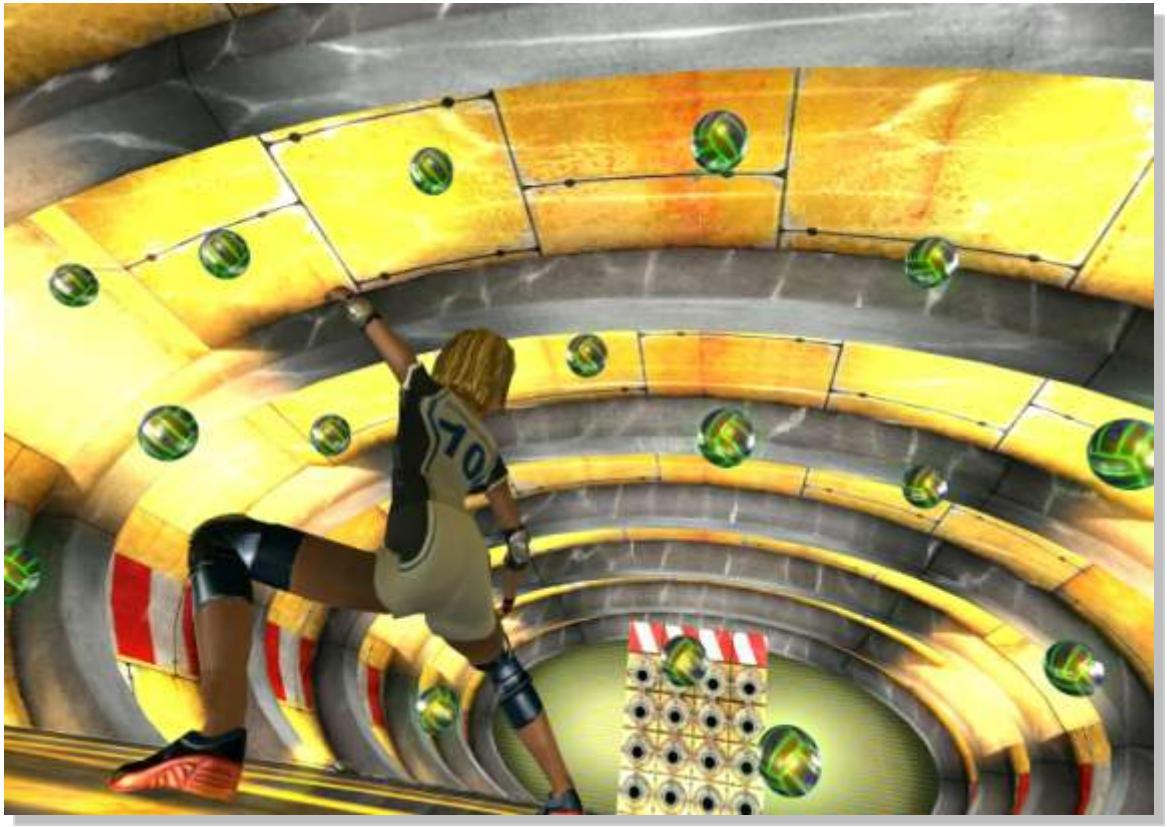
- **Layar**
  - Augmented reality browser for location based data
    - iPhone
    - Android

- **Kweekies**
  - Research project at int13 games

High Quality Graphics on Mobile

Imagination

- Advanced effects and complex scenes are possible on mobile
- How can these be used in AR applications?

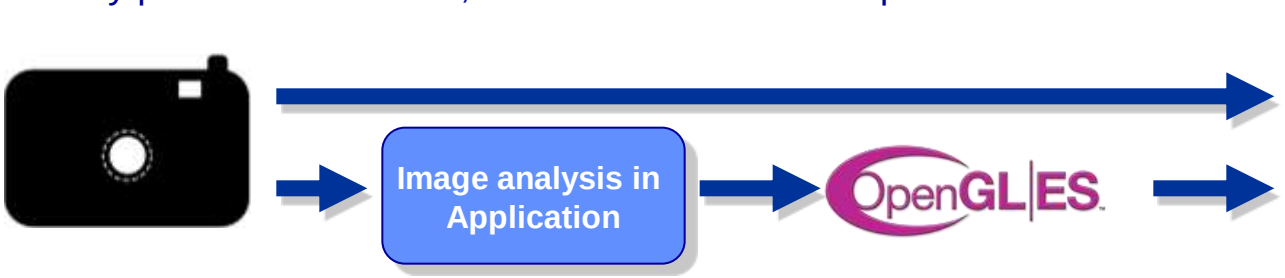


AR on Mobile
Naïve solution

- **Current implementation techniques are often inefficient**
  - Camera data is captured and copied into an application buffer
  - Image analysis is carried out here
  - The camera data is then uploaded to OpenGL ES and combined with 3D meshes
- **Benefits**
  - Easy and available on most platforms
- **Drawbacks**
  - Copying into application memory is **slow**
  - Uploading a texture each frame is **slow**
- **Bandwidth is precious on mobile systems**
  - Not much bandwidth left for graphics

AR on Mobile Overlay Solution

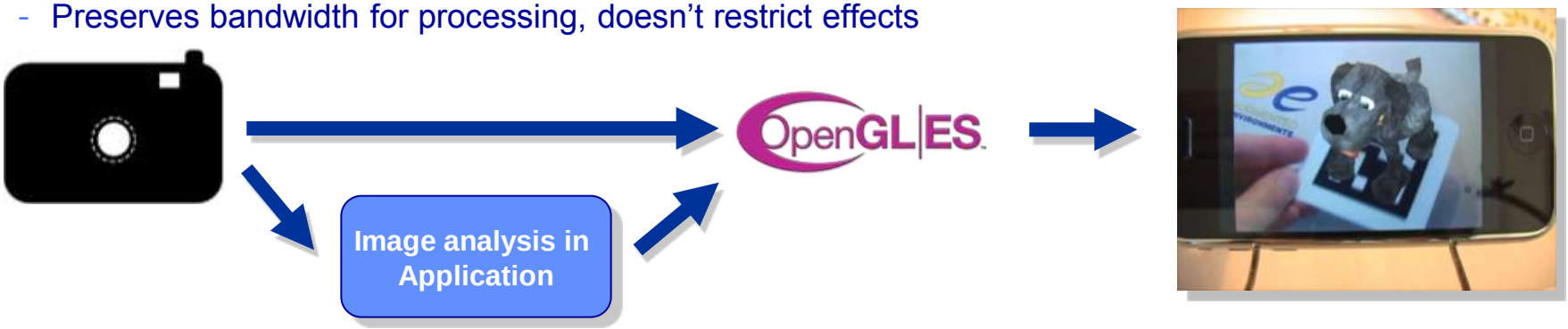
- **A better solution, but still not ideal**
  - Camera data is captured and sent straight to composition engine
  - Camera data is made available for application analysis
  - OpenGL ES content is rendered and overlayed by the composition hardware
- **Benefits**
  - Can be fast and is available on a number of platforms
- **Drawbacks**
  - There may still be copies in the implementation that are **slow**
  - There is no way to combine the camera data with generated graphics without uploading a texture each frame – and this is **slow**
- **Can be a fast solution, but inflexible**
  - May preserve bandwidth, restricts effects that are possible



AR on Mobile
Texture Streaming

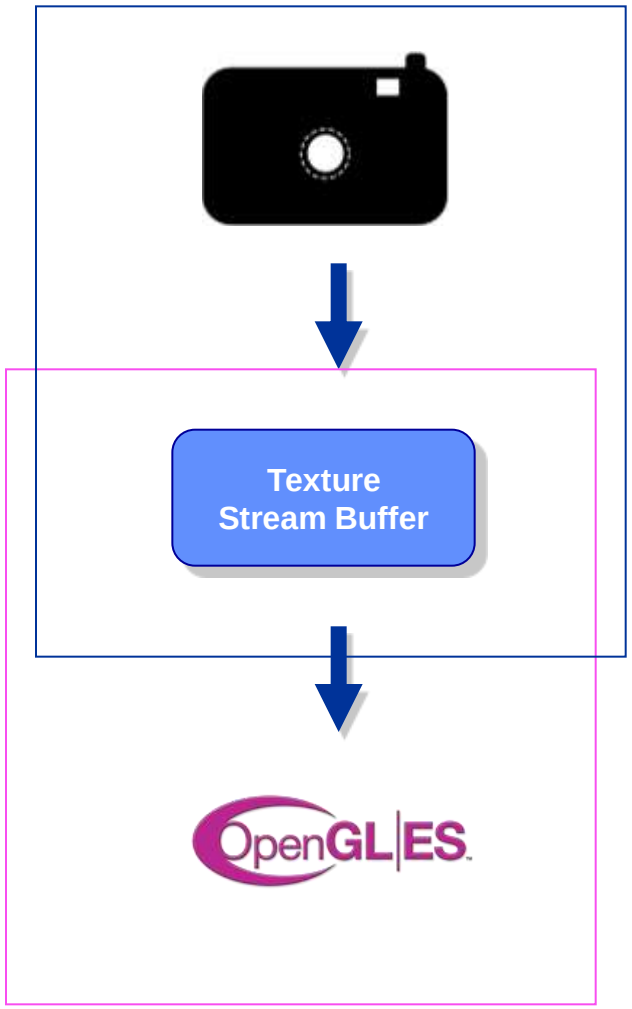
Imagination

- **Probably the best solution, if available**
  - Camera data is captured and shared with OpenGL ES in a buffer – no copy
  - Camera data is made available for application analysis from this buffer – no copy
  - All rendering is done through OpenGL ES, including camera data
- **Benefits**
  - Potentially fastest and most flexible method
  - More effects available
    - 3D meshes using the view for reflections, refractions etc.
    - Post-processing of camera data before display (hi-lights, image intensifier effect)
- **Drawbacks**
  - There may still be copies in poor implementations
  - Not available on many platforms
- **Fastest, most flexible solution, where available**
  - Preserves bandwidth for processing, doesn't restrict effects



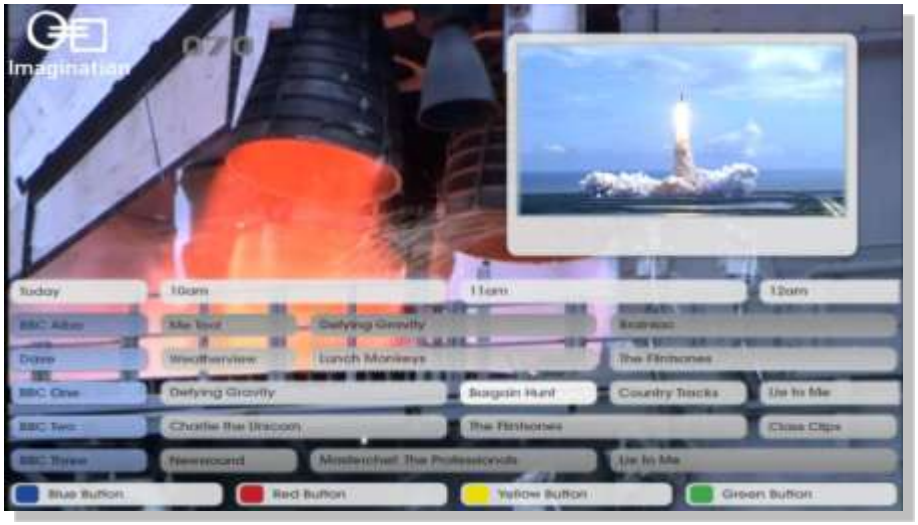
Texture streaming

- **What is texture streaming?**
  - The process of writing images into buffers that can be accessed directly by the graphics hardware
- **Benefits**
  - Fast image access (Zero-copy on UMA devices)
  - Multi-buffered
  - Can be used as a texture
- **Drawbacks**
  - Not available on many devices
  - Lacks a standard API
  - All buffers within a stream must be the same format and resolution
- **Hardware Support is there**
  - All Imagination POWERVR SGX hardware can support this
  - Similar solutions are possible for other hardware



Texture streaming: Examples

- **Set top box demo**
  - Texture streaming allows multiple video streams to be applied to 3D GUI elements
  - HD video in the background
  - All in real-time at high frame-rates



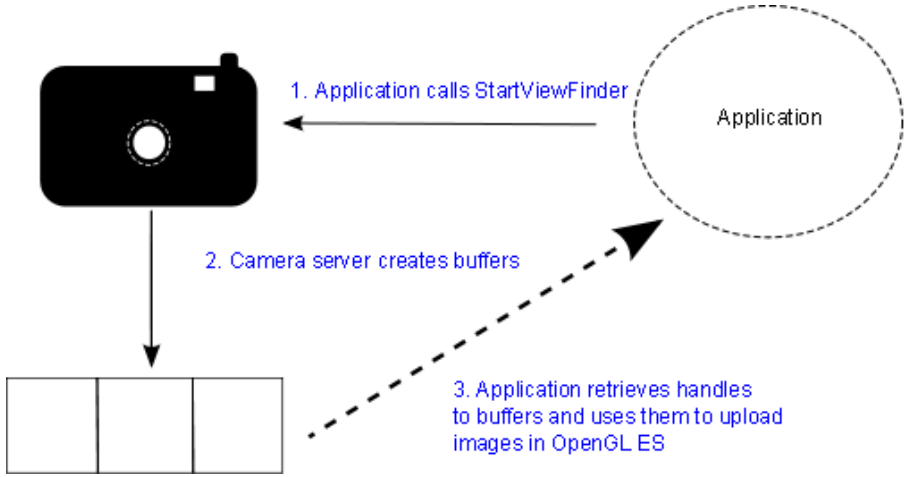
- **Video cube demo**
  - Frames from a video can be used to texture the faces of a cube object
  - Each face has a different post processing effect applied to it
  - All in real-time at high frame-rates

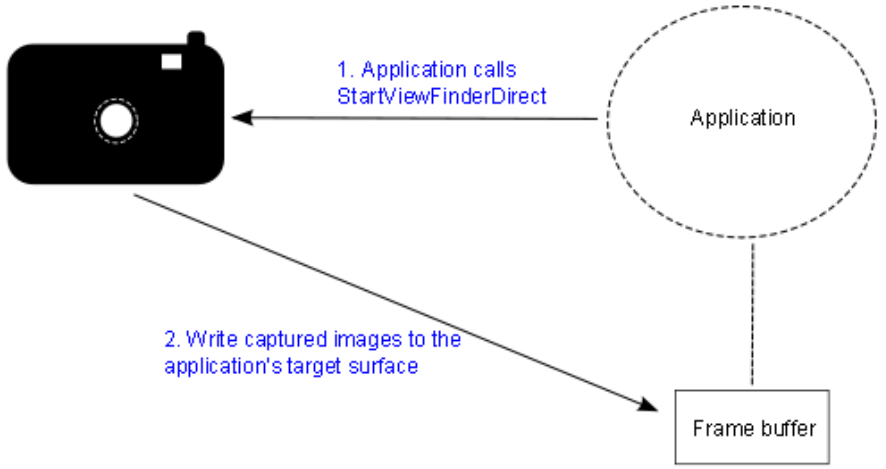
Sony-Ericsson Vivaz

- **Specs**
  - 360x640 pixel screen
  - POWERVR SGX 530
- **Why was this device used?**
  - One of the few currently shipping devices that provides texture streaming functionality

Capturing images on the Vivaz

- **Capturing images**
  - Once the camera server has been started, there are two default Symbian functions for beginning an image capture stream; StartViewFinderL & StartViewFinderDirectL
- **StartViewFinderL**
  - Store captured images in buffers allocated by the camera server
  - **Benefits**
    - Buffers can be mapped into user mode, where the CPU can analyze them
  - **Drawbacks**
    - Slow per-frame upload through graphics APIs
- **StartViewFinderDirectL**
  - Write captured images directly into the frame buffer
  - **Benefits**
    - Fast update of captured images
  - **Drawbacks**
    - Difficult to access the image for CPU analysis
    - Cannot be used as a texture in OpenGL ES

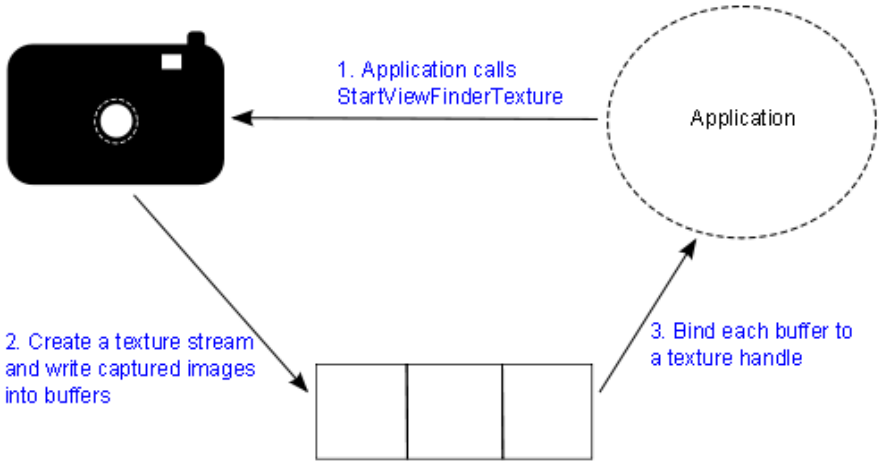




Vivaz Texture Streaming

- **Capturing images**
  - Sony-Ericsson's StartViewFinderTexture & StopViewFinderTexture functions can be accessed through a camera extension
- **StartViewFinderTexture**
  - Write captured images into a texture stream

  Benefits
  - Fast update of captured images
  - Images can be textured onto any 3D object
- **DynamicTexture.h**
  - Provides a user-mode API to the kernel-mode texture stream module
  - Consuming applications can use this to retrieve texture stream information



Texture streams in OpenGL ES 2.0

Imagination

- **Imagination's texture stream extensions**
  - void glGetTexStreamDeviceAttributeivIMG(GLint texStreamId, GLenum pname, GLint *params)
  - void glTexBindStreamIMG(GLint texStreamId, GLint bufferId)
  - const GLubyte* glGetTexStreamDeviceNameIMG(GLint texStreamId)

- **Loading textures**

```
glGenTextures(m_i32NoTexStreamBuffers, m_auiTexStreamTexId);

// Bind tex stream textures
glActiveTexture(GL_TEXTURESTREAM);
for(int i = 0 ; i < m_i32NoTexStreamBuffers; ++i)
{
    glBindTexture(GL_TEXTURE_STREAM_IMG, m_auiTexStreamTexId[i]);
    glTexParameterf(GL_TEXTURE_STREAM_IMG, GL_TEXTURE_MIN_FILTER, TEXTURE_STREAM_FILTER);
    glTexParameterf(GL_TEXTURE_STREAM_IMG, GL_TEXTURE_MAG_FILTER, TEXTURE_STREAM_FILTER);

    glTexBindStreamIMG(m_uiTexStreamId,i);
}
```

Texture streams in OpenGL ES 2.0

Imagination

- **Use a shader program**
  - Retrieve the ID of the most recently written texture stream buffer
  - Bind the corresponding texture

```
// Use shader program
glUseProgram(m_Tex2DShader.uiId);

// Retrieve the current read buffer number and use it
int i32BufferId = DynamicTexture::CurrentBuffer(bufferDevice);

// Bind the most recently captured image
glActiveTexture(GL_TEXTURESTREAM);
glBindTexture(GL_TEXTURE_STREAM_IMG, m_auiTexStreamTexId[i32BufferId]);

glUniformMatrix4fv(m_Tex2DShader.uiMVPMatrixLoc, 1, GL_FALSE, m_mProjection.ptr());

// Enable the vertex attribute arrays
glEnableVertexAttribArray(VERTEX_ARRAY);
glEnableVertexAttribArray(TEXCOORD_ARRAY);

glVertexAttribPointer(VERTEX_ARRAY, 4, GL_FLOAT, GL_FALSE, 0, &vVertices);
glVertexAttribPointer(TEXCOORD_ARRAY, 2, GL_FLOAT, GL_FALSE, 0, &vTexCoords);

// Draw primitive
glDrawArrays(GL_TRIANGLE_FAN, 0, 4);

// Safely disable the vertex attribute arrays
glDisableVertexAttribArray(VERTEX_ARRAY);
glDisableVertexAttribArray(TEXCOORD_ARRAY);
```

Texture streams in GLSL ES

- **Sample texture stream texture**
  - Enable the GL_IMG_texture_stream2 extension if it exists
  - Indicate that the texture stream sampler (samplerStreamIMG) should be used
  - Use the texture stream sampler to retrieve the texel colour at the specified texture coordinate

```
#ifdef GL_IMG_texture_stream2
#extension GL_IMG_texture_stream2 : enable
#endif

#ifdef STANDARD_TEXTURE
    uniform sampler2D texStream;
#else
    uniform samplerStreamIMG texStream;
#endif

varying highp vec2 TexCoord;

void main()
{
    #ifdef STANDARD_TEXTURE
        gl_FragColor = texture2D(texStream, TexCoord);
    #else
        gl_FragColor = textureStreamIMG(texStream, TexCoord);
    #endif
}
```

Augmented reality demo

- **High frame rate refraction demo**
- **>20,000 polygons in the scene (can support more)**

- **Refraction effect requires texture data to be available to OpenGL ES**
  - Without texture streaming this runs much more slowly

Future of Mobile AR

- **Faster CPUs and GPUs**
  - Better image analysis and graphics
  - Dependency on markers may be eliminated
- **Better cameras and other sensors**
  - More accuracy in interaction
  - Higher video frame rates and resolution
- **More applications**
  - Multiplayer alternative world games
  - Vehicle HUDs

- **GP-GPU image analysis**
  - OpenCL enabled mobile devices are likely to ship within the next year
    - On GPU image analysis

Where to get the POWERVR SDK, Documentation & Support

- **Available through the POWERVR *Insider* Programme**
  - Click on “Developers” on the Imagination website: http://www.imgtec.com
  - Free to join!
- **Benefits of being a POWERVR *Insider***
  - Access to the SDK downloads
  - Documentation – performance guidelines, explanations of POWERVR technologies and utility user guides
  - FAQs
  - Developer forums
  - Direct email contact to POWERVR Developer Technology: devtech@imgtec.com
  - Partner Program
  - Newsletter
  - Cross-promotions through press announcements and print/online media
  - Tradeshow Partnership

Imagination
TECHNOLOGIES

Questions?

Imagination
TECHNOLOGIES

Augmented reality on mobile platforms

POWERVR Developer Technology

Contact: devtech@imgtec.com